data structure defining a nested, hierarchial relationship such that said field objects are encapsulated within said record objects and wherein said record objects encapsulated within said update objects;

said update object thereby being configured to automatically store data and to automatically store in said attributes an historic record of processes performed on said data as said update object is routed anywhere throughout said communication network,

said computer system further generating a database system that comprises a collection of record objects that define a persistent portion of the database and a collection of update objects that define a transient portion of the database;

the database system being responsive to said processing tag for automatically updating the persistent portion of said database;

the database system further including a feedback mechanism comprising a second update object for systematically purging said update objects from the transient portion of the database.

65. The database system of claim **64** wherein said record object includes a field object for storing said element of information.

66. The database system of claim **64** wherein said update object includes an acknowledgement tag accessed by said feedback mechanism in purging said update objects from the transient portion of the database.

67. The database system of claim **64** wherein said processing tag comprises a routing tag and a database modification tag.

68. The database system of claim **64** wherein said record object includes a record identification tag.

69. The database system of claim **64** wherein at least one of said record objects and said update objects employs a variable length data storage mechanism.

70. The database system of claim **69** wherein said variable length storage mechanism uses a pointer system for data storage and retrieval.

71. The database system of claim **64** wherein said database system includes a first database having a first processor and wherein said feedback mechanism uses said first processor to read the processing tag of said update object and to store said update object in the transient portion of the first database.

72. The database system of claim **64** wherein said database system includes a first database having a first processor and a second database having a second processor and wherein said feedback mechanism uses said first processor to read the processing tag of said update object and to route said update object to said second database in accordance with said processing tag.

73. The database system of claim **64** wherein said database system includes a first database having a first processor and a second database having a second processor and wherein said feedback mechanism uses said second processor to generate a second update object having a second processing tag.

74. The database system of claim **73** wherein said second processing tag is an acknowledgement tag.

75. The database system of claim **64** wherein said database system includes a first database having a first processor and a second database having a second processor and wherein said feedback mechanism uses said second processor to transmit a second update object having second processing tag to said first processor.

76. The database system of claim **64** wherein said database system includes a first database having a first processor and a second database having a second processor;

wherein said feedback mechanism uses said first processor to read the processing tag of a first update object and to store said first update object in the transient portion of the first database;

wherein said feedback mechanism uses said first processor to transmit said first update object to said second processor;

wherein said feedback mechanism uses said second processor to transmit a second update object having a second processing tag to said first processor in response to said first update object; and

wherein said feedback mechanism uses said first processor to read said second processing tag and to modify said first database by deleting said first update object from said first database in response to said second processing tag.

* * * * *